5. Installatie van Python en MicroPython

Python is een programmeertaal die begin jaren 90 ontworpen en ontwikkeld werd door Guido van Rossum van het Centrum voor Wiskunde en Informatica in Amsterdam. De ontwikkeling van de taal wordt nu geleid door de Python Software Foundation. Python is vrije software. Python heeft zijn naam te danken aan het favoriete televisieprogramma van Guido van Rossum, Monty Python's Flying Circus.

Python is op korte tijd heel populair geworden. Veel hogescholen en universiteiten onderwijzen Python als eerste computertaal. Waarom?

- Python is eenvoudig aan te leren. Python programma's vallen op door compactheid en helderheid. Bovendien zijn er geen puntkomma's en accolades nodig, talen zoals C staan er bol van.
- Python is een heel krachtige taal, er zijn heel veel softwarebibliotheken beschikbaar.
- Je kunt met Python interactief werken. Je geeft een opdracht en je krijgt direct een resultaat. Dat is ideaal om snel iets uit te proberen.
- Voor talen zoals C, C++, … bestaan er **compilers** die de broncode vertalen naar machinetaal die de computer zo kan uitvoeren. Een Python programma wordt niet gecompileerd, de **interpreter** zal de instructies van het programma één voor één uitlezen en direct uitvoeren. Het voordeel is dat we een programma heel snel kunnen wijzigen. Dat heeft ook nadelen: een Python programma is langzamer dan een gecompileerd programma. Bovendien moeten moet de interpreter op het doelsystem geïnstalleerd zijn. We hebben dus krachtigere controllers nodig maar die zijn tegenwoordig goed verkrijgbaar en veel kosten ze ook niet meer.
- Python is object georiënteerd, zie verder in dit boek.

MicroPython is een implementatie van Python en is geoptimaliseerd om op microcontrollers te werken. Microcontrollers hebben veel minder geheugen dan Pc's. Net zoals bij Python wordt de code niet gecompileerd, maar tijdens het uitvoeren geïnterpreteerd. Je vind MicroPython op www.micropython.org .

MicroPython is board-specifiek en controller-specifiek. Een versie voor controller A is niet compatibel met die voor controller B. Voor elke controller bestaan er verschillende controller bordjes en MicroPython probeert voor veel bordjes een eigen versie van de software te ontwikkelen. Op de downloadpagina van MicroPython zie je een overzicht. Heb je een bordje dat niet in die lijst staat, probeer dan de software van een bordje met dezelfde controller, dat werkt waarschijnlijk.

*De systeemsoftware voor MicroPython op de downloadpagina is een bestand met extensie .uf2 of .bin of nog iets anders. **Gebruik het .uf2-bestand**, tenzij je hieronder de .bin nodig hebt.*

Hoe werkt MicroPython op een RP2040? De bootloader in de ROM van de RP 2040 doet enkele controles bij het opstarten. Als knop BOOTSEL is ingedrukt bij het opstarten of als er geen software in het FLASH geheugen zit, dan gedraagt de RP2040 zich als een schijfstation van de PC. We kunnen dan software overbrengen naar het FLASH geheugen. In het andere geval wordt de software in de FLASH uitgevoerd. Als de bootloader een interpreter vindt en Python programma main.py, dan start main.py op. Staat er geen main.py in de FLASH, dan start de interpreter op in interactieve modus. We hebben dus software nodig op de PC om programma's in te voeren en software om programma's over te zeten naar de controller. Voor die taken gebruiken we programma Thonny.

Thonny

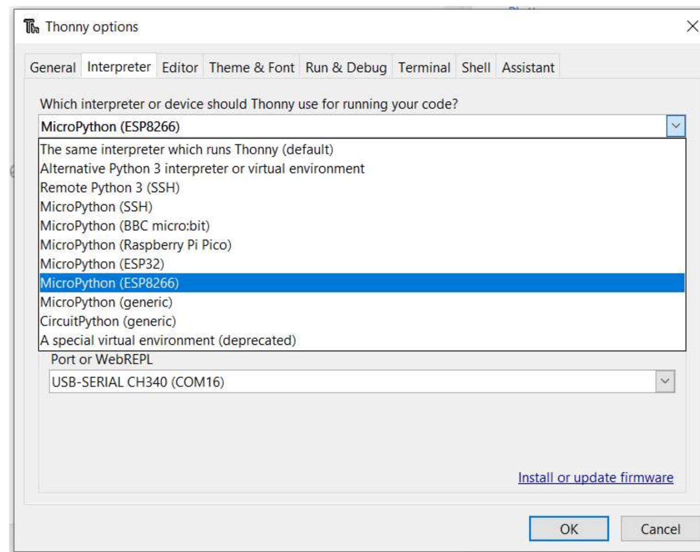
Thonny is een gratis en open-source **geïntegreerde ontwikkelomgeving** voor Python. Het is gemaakt door Aivar Annamaa van de universiteit van Tartu in Estland. Je vindt het programma op https://thonny.org . Bovenaan die webpagina zie je een link naar de downloads, o.a. voor Windows. Download de Windows installer en voer die uit. Je krijgt waarschijnlijk enkele waarschuwingen, o.a. van de virusscanner. Negeer die.

Installatie van MicroPython door Thonny

Download eerst de meest recente versie van MicroPython voor jouw systeem. Die vind je op de downloadpagina van micropython.org. Op die pagina staat veel informatie, kies eerst de MCU (de controller) van jouw systeem. Heb je een RP2040, dan vind je een heleboel bordjes met een RP2040. De versie voor de Raspberry Pi Pico werkt met alle RP2040 bordjes. Heb je een ESP8266 of een ESP32, dan staat jouw systeem waarschijnlijk niet in de lijst. Kies dan ESP32/ESP8266 Espressif. Let op, er bestaan Releases en Nightly Builds. Een **Nightly Build** is een voorlopige, niet stabiele versie. Een **Release** is een stabiele versie. Kies de Release met het hoogste versienummer. Na downloaden staat de software in folder Downloads van de PC.

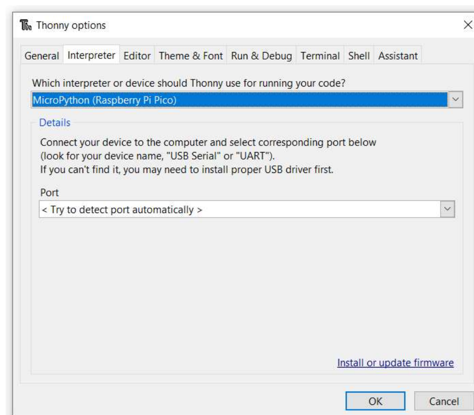
Je installeert MicroPython door de interpreter te installeren in het Flash-geheugen van de controller. Dat kan met Thonny: start Thonny op en sluit het bordje aan op een USB-poort van de PC. Ga naar menu Tools - Options – Select Interpreter en kies de *MicroPytho-versie voor jouw systeem*.

Deze installatiemethode werkt alleen als Thonny de controller hiervoor ondersteund. De ondersteunde controllers vind je bij Thonny in Menu Tools – Options – Interpreter.

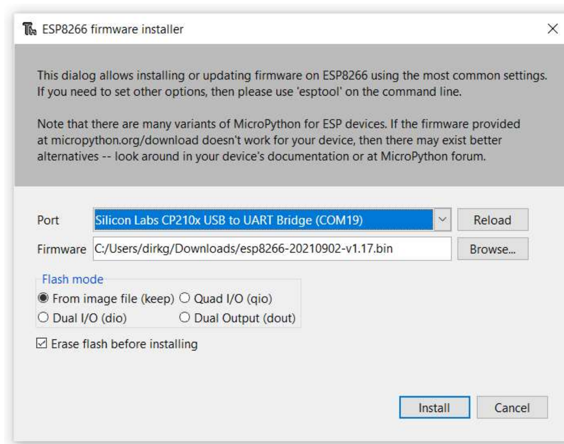


Figuur 20: ondersteunde controllers

Klik dan onderaan op *Install or Update firmware* om de interpreter te installeren in de controller.



Figuur 21: configuratie van Thonny



Figuur 22: installatie van de software

Stel de juiste Poort in. De automatische detectie van de poort werkt niet altijd even vlot. Klik op het keuzeknopje naast Port zonder aangesloten controller en ga na welke poortnummers beschikbaar zijn. Sluit dan de controller aan, open de options en ga terug kijken bij Port. De extra poort in dat scherm is de poort voor de jouw bordje, selecteer die. Laat de andere instellingen ongewijzigd en druk op Install. Even later zijn Thonny en MicroPython klaar voor gebruik.

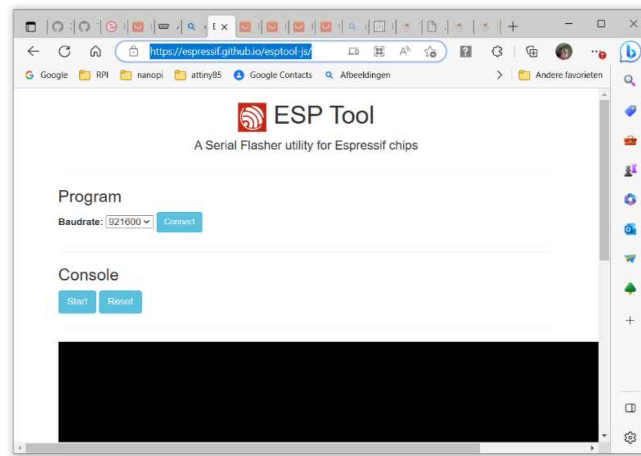
Kies de juiste USB-poort en blader naar het gewenste firmware-bestand. Voor een esp8266/esp8285 stel je de **flash mode** in als **Dual I/0**, voor de andere kies je **From image file**.

Installatie van MicroPython door kopiëren

Sommige controllers kunnen zich voordoen als een extra schijfeenheid. Een RP2040 breng je in die toestand door hem op te starten of te resetten met ingedrukte BOOT-toets. Voor andere, zoals de SAMD21 moet je tweemaal de resettoets indrukken. Als de controller in die toestand staat, dan kan je de software er gewoon in kopiëren. Na een reset is MicroPython klaar voor het gebruik.

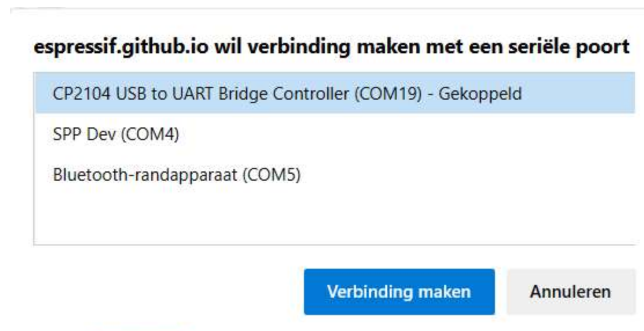
Installatie op een ESP-controller

Op de website van Espressif vind je een online tool om ESP-controllers te programmeren. Download de MicroPython-versie voor jouw controller en start de tool, zie https://espressif.github.io/esptool-js



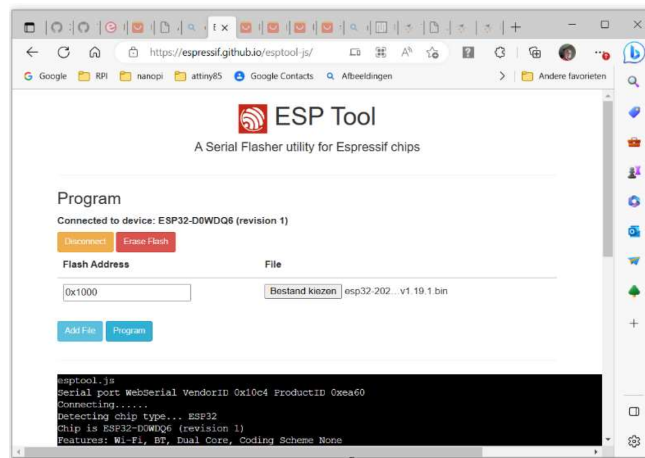
Figuur 23: ESP-tool

- Sluit de controller aan en klik op Connect



Figuur 24: ESP-tool, verbinding maken

- Selecteer de USB-poort van de controller. (COM19 in het voorbeeld) en klik op verbinding maken.



Figuur 25: ESP-tool, bestand kiezen

- Klik op kies bestand, kies de software die je gedownload hebt voor de controller en klik op program. Na enkele minuten is de controller klaar. Reset hem door even los te koppelen.

De instructies voor andere controllers vind je op de downloadpagina's van micropython.org.

Installatie van Thonny op niet-Windows toestel

Thonny werkt ook op Macintosh, op een Raspberry Pi 4 computer en op een Linux-systeem. Op thonny.org vind je meer informatie.

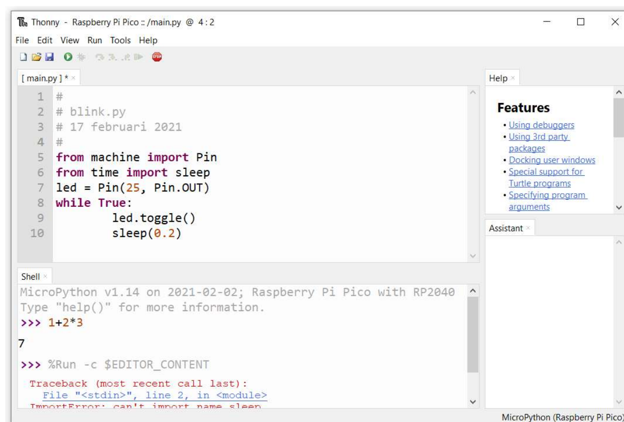
Gebruik van Thonny

Het **shell-venster** is het grote vak onderaan links. Hier kan je Python commando's geven. Die worden uitgevoerd door de controller. Typ een opdracht en je krijgt direct een resultaat. Merk op dat Python de wiskundig juiste volgorde van bewerkingen gebruikt.



Figuur 26: Thonny interactief

In het bovenste venster is het **editor-venster**. Hier typ je een programma in. De editor herkent de meest voorkomende typfouten, zoals ontbrekende haakjes en aanhalingstekens, en markeert die met een kleur. *Typ het programma van het voorbeeld hieronder in.* Je start het programma op met de F5-toets of met het start-pictogram, het groene knopje van de werkbalk. Thonny vraagt waar je het programma wilt bewaren. Dat kan op de PC of op de controller. Bewaar het op de PC. Als alles goed gaat zal het programma de LED van het bordje laten knipperen. Wil je de controller stand-alone gebruiken, dan moet je het programma bewaren op de controller met naam **main.py**, het start dan automatisch op bij het opstarten van de hardware.



Figuur 27: programma blink.py

Een programma moet bewaard worden voor we het kunnen uitvoeren.

Als er bij de uitvoering fouten optreden, dan geeft de interpreter een foutmelding. Soms is de fout eenvoudig te herstellen. Indien niet, kan je de **debugger** gebruiken. Die start je met Contr-F5. De debugger voert het programma regel per regel uit. Daarbij verschijnen in kleurige venstertjes de variabelenamen met hun inhoud, resultaten van expressies en van functieaanroepen.

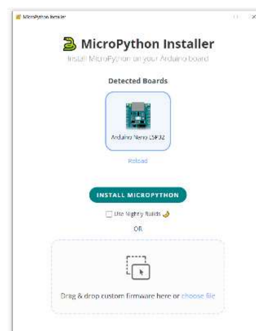
Arduino software voor MicroPython

Arduino en MicroPython? Nee, Arduino heeft geen plannen om zijn programmeer-taal te vervangen door MicroPython. Voor al hun boards geven zij volledige Arduino-ondersteuning. Enkele van hun boards zijn geschikt voor MicroPython en MicroPython levert er ook de nodige systeemsoftware voor. Arduino wil dit ondersteunen met eigen **installer** en een eigen **geïntegreerde ontwikkelomgeving**

De Arduino MicroPython Installer

Arduino heeft een programma ontwikkeld waarmee je op een eenvoudige manier de MicroPython firmware kunt installeren op jouw bordje. *Helaas, dit programma werkt alleen met Arduino bordjes, niet op bordjes van andere fabrikanten.* We hebben het programma geteste met een Nano ESP32.

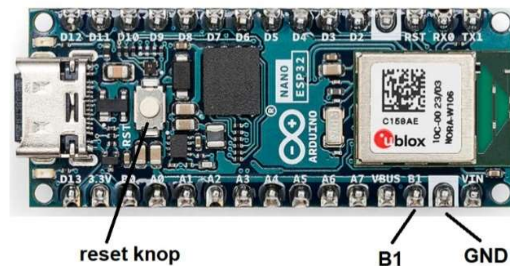
Je vindt het programma op de website van Arduino. Ga naar **Software** en klik op de link in paragraaf **MicroPython with Arduino Boards**. Op die pagina zie je een blokje **Board installer**. Volg die link en download de **Arduino MicroPython Installer**. Sluit het board aan op een USB-poort. De download is een .exe-bestand. Start die op.



Figuur 28: MicroPython Installer

Het programma vindt direct de aangesloten Arduino Boards. De eenvoudigste manier om de firmware, de MicroPython interpreter, op het bordje te krijgen is knop **Install**. Het programma zoekt op Internet de nieuwste versie van de firmware en installeert die.

Onze Nano ESP32 gaf foutmeldingen tijdens de testen. Wij hebben dit opgelost door het bord op te starten in de **Firmware Download mode**: verbind **GND** met **B1** en druk op de **Reset**-knop. Verbreek dan de verbinding tussen **GND** en **B1**. De RGB-led zal geel of paars oplichten. Probeer nu te installeren.

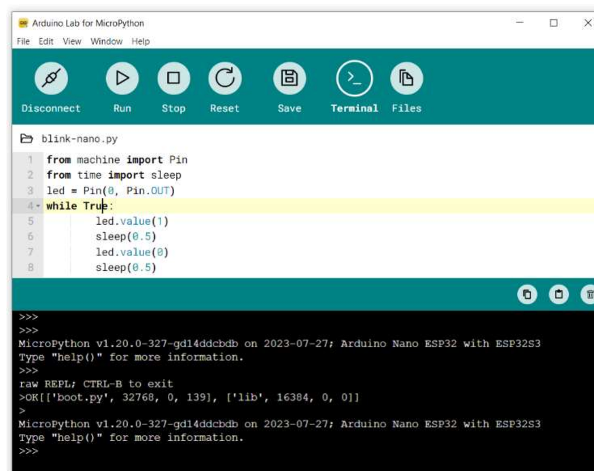


Figuur 29: Firmware Download Mode

Arduino Lab for MicroPython

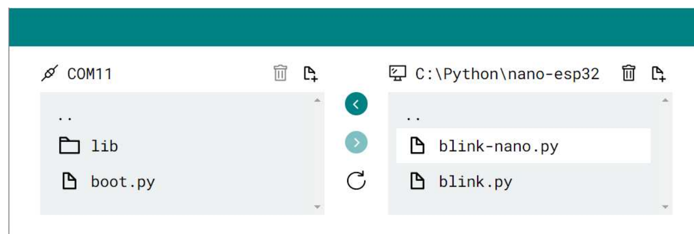
Met dit programma ontwikkel je MicroPython scripts en stuur je ze door naar het ontwikkelbord. *Dit programma werkt met alle bordjes met MicroPython firmware, dus ook met niet-Arduino bordjes.* Je vindt het op de Arduino website, op dezelfde plaats als de installer. Je kunt het downloaden als een .zip-bestand. Bewaar het in jouw werkmap en pak alles uit.

Na uitpakken vind je een folder Arduino.lab.for.MicroPython-win_x64 met daarbinnen een .exe bestand. Start dat op.



Figuur 30: Arduino Lab for MicroPython

- De menubalk (File, Edit, …) heeft opties die je in bijna alle Windows programma's vindt
- De knoppen van de knoppenbalk zijn:
    - **Connect/Disconnect**: hiermee maak je verbinding met een MicroPython-bordje. *Arduino-Lab ondersteunt alle bordjes, ook niet-Arduino bordjes*.
    - **Run** voert de programma-code in het editorvenster uit.
    - **Stop** onderbreekt het programma.
    - **Save** bewaart de programmatekst. De bestandsnaam kies je zelf. Hij staat boven de editor.
    - **Terminal** opent of sluit het zwarte terminalvenster onderaan. Hier vind je uitvoer van de controller bij opstarten of uitvoer van het programma.
    - **Files** opent of sluit het bestandsvenster. Links zie je de bestanden en folders in de controller, rechts die van de PC. Met de groene knopjes kopieer je bestanden van de ene naar de ander kant.



Figuur 31: het bestandsvenster

Opmerkingen

- De controller heeft een folder **lib**, zet daar de libraries voor jouw programma.
- Arduino-Lab zal de RTC van de controller **niet** instellen. Andere programmeeromgevingen zoals Thonny doen dit wel!